**ANALYSIS ON THE SELLING PRICE OF USED CARS**

**USING PYTHON**

**INTRODUCTION:**

The used car market is a dynamic and rapidly evolving segment of the automotive industry. With growing consumer interest in pre-owned vehicles, understanding the factors that influence their selling prices has become increasingly important for buyers, sellers, and market analysts alike. This project aims to analyse the selling price of used cars using Python, leveraging data analytics and machine learning techniques to uncover trends and patterns in the market.

The primary objective of this analysis is to identify key factors that drive price variations in used cars, such as vehicle age, mileage, brand, model, fuel type, transmission type, and geographical location. By exploring these variables, we aim to provide actionable insights that can assist stakeholders in making informed decisions.

Using Python's robust data processing and visualization libraries, we will:

**Preprocess the Data**: Clean and prepare the dataset by handling missing values, outliers, and categorical variables.

**Exploratory Data Analysis (EDA)**: Investigate the dataset to identify patterns, correlations, and trends.

**Modeling**: Build predictive models to estimate the selling price based on input features.

**Insights and Recommendations**: Interpret the results to provide meaningful conclusions for buyers and sellers.

This project not only demonstrates the power of Python in data analysis but also sheds light on the complexities of pricing in the used car market, helping participants understand market dynamics and make data-driven decisions.

**OBJECTIVE:**

The objective of this project is to analyze the factors influencing the selling price of used cars using Python and statistical modeling. By leveraging data analytics techniques, the project aims to:

**Identify Key Determinants:** Understand how variables such as vehicle age, mileage, brand, model, fuel type, and transmission impact the selling price.

**Uncover Market Trends:** Detect patterns and trends in the used car market, including pricing differences across regions and time periods.

**Build Predictive Models:** Develop machine learning models to accurately predict the selling price of used cars based on their features.

**Provide Data-Driven Insights:** Generate actionable insights that can assist buyers and sellers in making informed decisions.

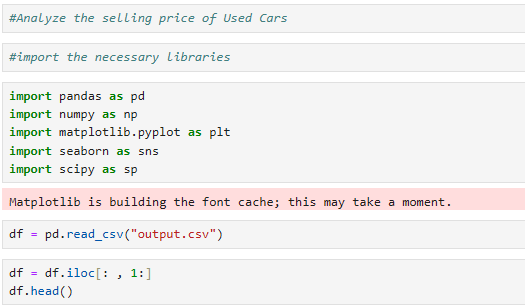
**Libraries used:**

1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. Scipy

**TASK:**

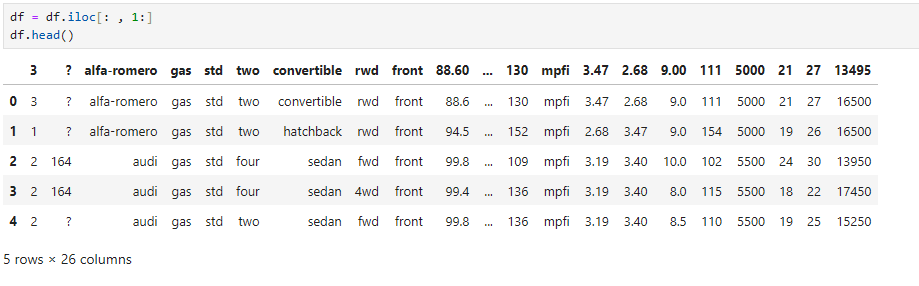
1. IMPORT THE DATASET
2. OBSERVANCE
3. DEFINE HEADERS
4. ANALYSIS ON THE MISSING VALUES
5. CONVERSION
6. DATATYPE CONVERSION
7. NORMALIZING VALUES
8. DESCRIPTIVE ANALYSIS
9. VISUALIZATION
10. GROUPING
11. PIVOT METHOD
12. RESULT
13. IMPORT THE DATASET:

* We use a CSV file for analysis. The dataset is uploaded to the Jupyter notebook. The CSV file is called out using the pandas library.
* All the necessary libraries required for analysis and visualization and called out.



1. OBSERVANCE:

* Let’s understand our dataset now. We could possibly call out header function to view the first 5 rows to observe the format of the dataset.



* Our dataset holds 26 columns in total.

1. DEFINE HEADERS

* We have an option to rename the headers of the columns. On observing the dataset, the column names consists of numbers and short-forms. Let’s rename them by columns header function.
* headers = ["symboling", "normalized-losses", "make",

"fuel-type", "aspiration","num-of-doors",

"body-style","drive-wheels", "engine-location",

"wheel-base","length", "width","height", "curb-weight",

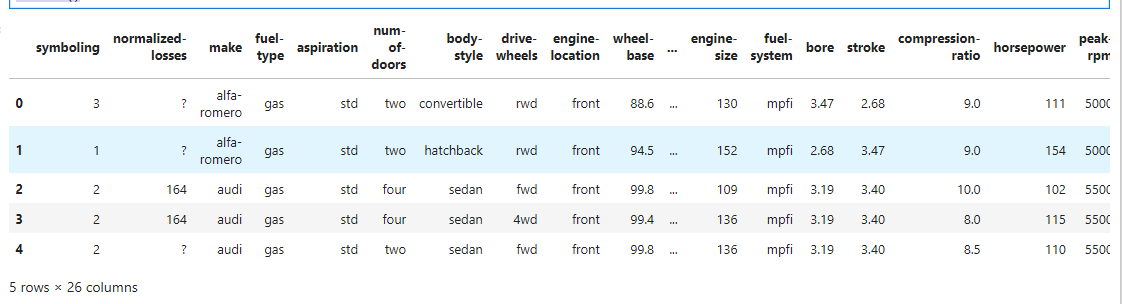
"engine-type","num-of-cylinders", "engine-size",

"fuel-system","bore","stroke", "compression-ratio",

"horsepower", "peak-rpm", "city-mpg", "highway-mpg", "price"]

df.columns = headers

df.head()

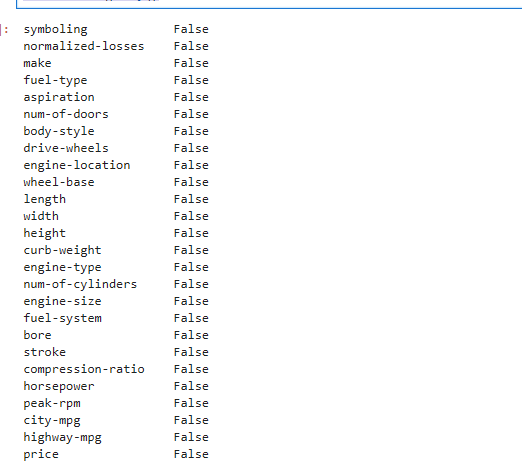


1. ANALYSIS ON THE MISSING VALUES:

* Our analysis primarily should include filtering out the missing or null values. These might cause distortion in the results of the analysis. To be more precise, we could possibly filter out such situations.
* data = df

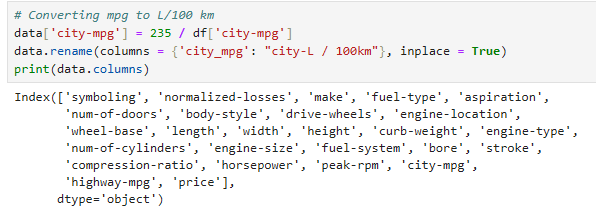
data.isna().any()

data.isnull().any()



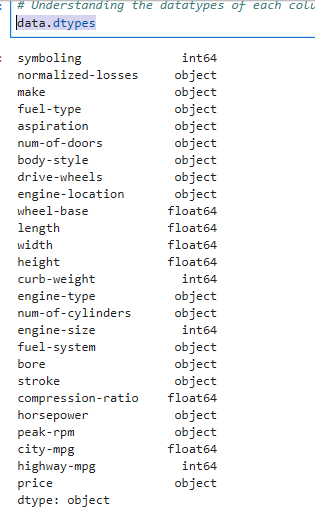
1. CONVERSION:

* We primarily perform conversion is to bring all the values to a common units. This elimination additional calculations and helps us to visualize the results better.



1. DATATYPE CONVERSION:

* Datatype conversion is required to change the object of certain columns. During visualization, it’s not advised to work with objects over integers or numeric values. We’ll perform the necessary datatype conversions.

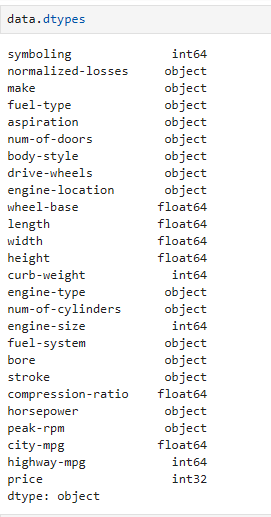


* The price columns should be an integer but here it is object, it’s because of the ‘?’ symbol present in one of the field.
* We’ll remove the ‘?’ symbol as it isn’t required and change the datatype to an integer

data.price.unique()

data = data[data.price != '?']

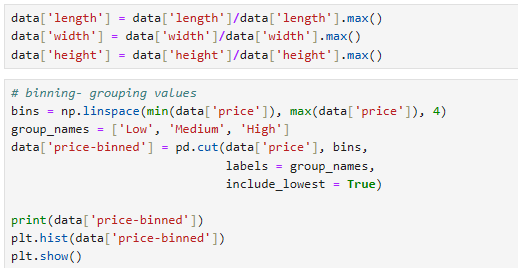
data['price'] = data['price'].astype('int')

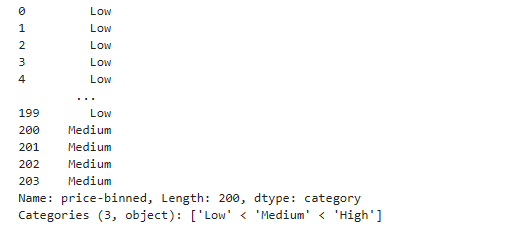


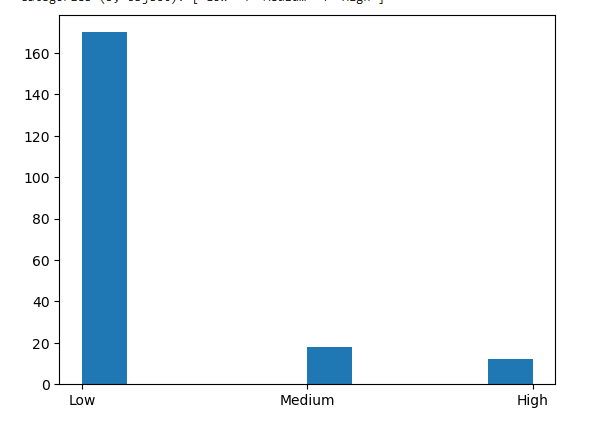
* The datatype of price is now changed to integer.

1. NORMALIZING VALUES:

* Normalizing values by using simple feature scaling method and binning- grouping values
* Normalization is the process of adjusting data values to fit within a specific range, such as [0, 1]. This is often done using the simple feature scaling method
* Binning involves dividing data into discrete intervals or "bins." It is used for simplification and categorical analysis, where continuous variables are converted into categories.







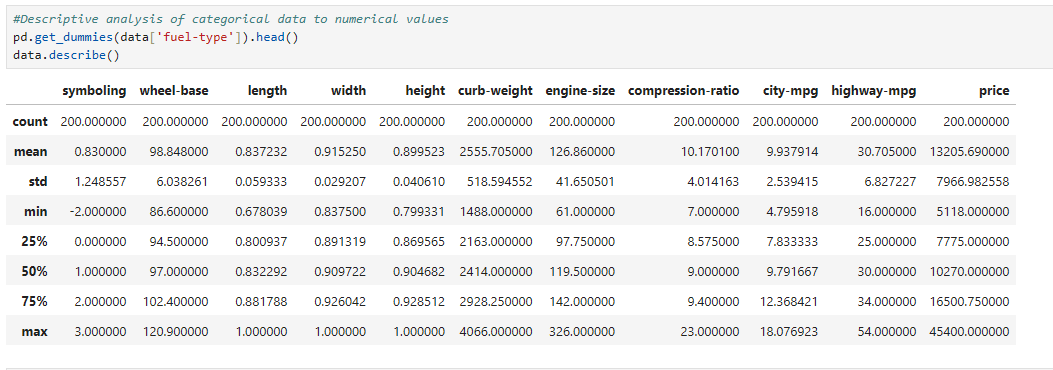
1. DESCRIPTIVE ANALYSIS:

* Doing descriptive analysis of data categorical to numerical values. By them we could achieve the following:

1. Facilitates Mathematical Operations
2. Enhances Compatibility with Models
3. Improves Data Visualization
4. Supports Statistical Analysis
5. Standardization and Consistency

* Benefits for Descriptive Analysis:

1. Identifying Trends: Numerical data enables better identification of relationships and patterns.
2. Enabling Advanced Techniques: Makes it possible to apply clustering, regression, and other numerical methods.
3. Quantifying Relationships: Helps in understanding the contribution or effect of different categories on the target variable.



1. VISUALIZATION:

* Visualization provides a quick and intuitive understanding of patterns, trends, and outliers in data.
* Let’s plot the data according to the price based on engine size.
* We use scatter plot to achieve the above task.

# examples of box plot

plt.boxplot(data['price'])

# by using seaborn

sns.boxplot(x ='drive-wheels', y ='price', data = data)

# Predicting price based on engine size

# Known on x and predictable on y

plt.scatter(data['engine-size'], data['price'])

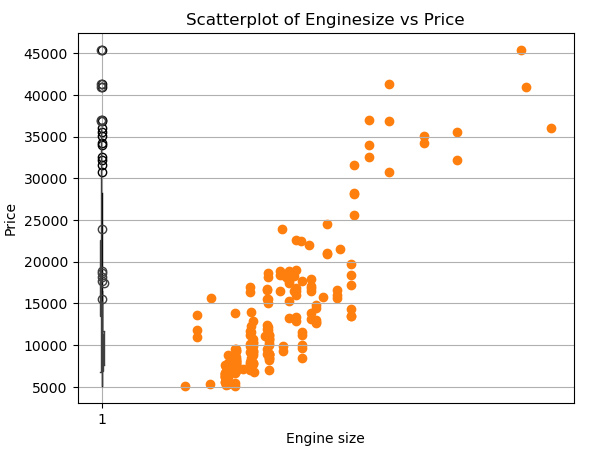
plt.title('Scatterplot of Enginesize vs Price')

plt.xlabel('Engine size')

plt.ylabel('Price')

plt.grid()

plt.show()



1. GROUPING:

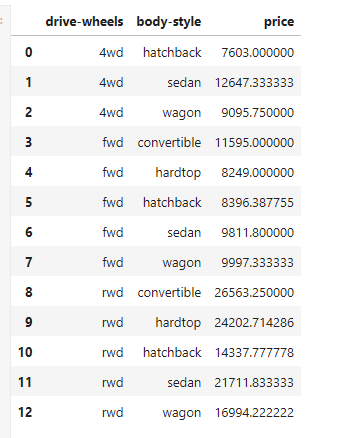
* Grouping enables us to analyze subsets of data by applying aggregate, transformation, or filtering operations to groups based on common characteristics.

test = data[['drive-wheels', 'body-style', 'price']]

data\_grp = test.groupby(['drive-wheels', 'body-style'],

as\_index = False).mean()

data\_grp



1. PIVOT METHOD:

* A Heat map visualizes data with various levels with the intensity using metrics. We derive the values using pivot method for better accuracy.

# pivot method

data\_pivot = data\_grp.pivot(index = 'drive-wheels',

columns = 'body-style')

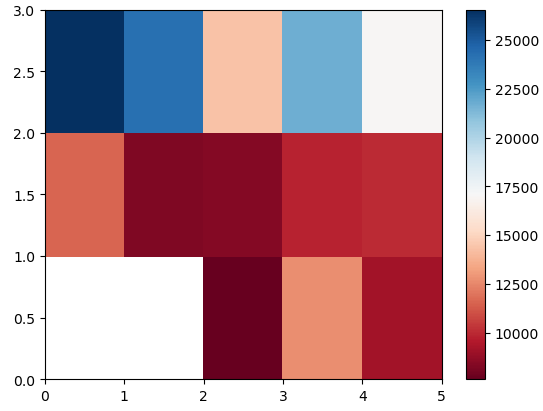
data\_pivot

# heatmap for visualizing data

plt.pcolor(data\_pivot, cmap ='RdBu')

plt.colorbar()

plt.show()



1. RESULT:

* Obtaining the final result and showing it in the form of a graph. As the slope is increasing in a positive direction, it is a positive linear relationship.
* A positive slope indicates that as the value of the independent variable (x-axis) increases, the value of the dependent variable (y-axis) also increases.
* This linear relationship is often represented by a straight line with a positive gradient in a scatter plot or line graph.

# Analysis of Variance- ANOVA

# returns f-test and p-value

# f-test = variance between sample group means divided by

# variation within sample group

# p-value = confidence degree

data\_annova = data[['make', 'price']]

grouped\_annova = data\_annova.groupby(['make'])

annova\_results\_l = sp.stats.f\_oneway(

grouped\_annova.get\_group('honda')['price'],

grouped\_annova.get\_group('subaru')['price']

)

print(annova\_results\_l)

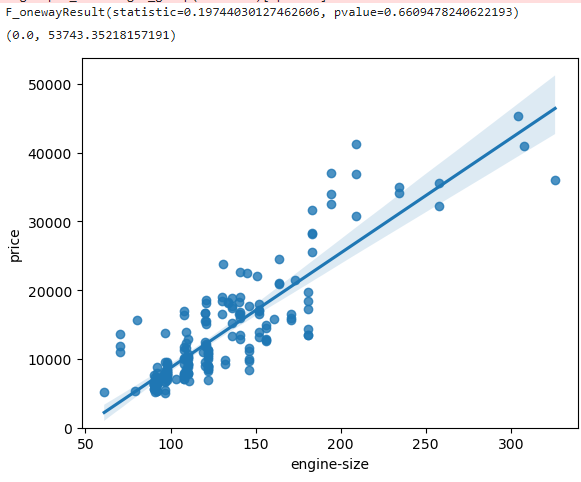
# strong corealtion between a categorical variable

# if annova test gives large f-test and small p-value

# Correlation- measures dependency, not causation

sns.regplot(x ='engine-size', y ='price', data = data)

plt.ylim(0, )



1. CONCLUSION:

In this project, we analyzed the selling price of used cars using Python, employing a combination of data preprocessing, exploratory data analysis, feature engineering. The primary objective was to identify the factors influencing the selling price of used cars and to predict future prices with accuracy.